LINKgroup

Distance Learning System

Upotreba biblioteka NumPy i matplotlib

Python Data Access

NumPy

- NumPy je Python biblioteka za napredno rukovanje nizovima / matricama
- Neizostavni je sastojak paketa za rad sa podacima

pip install numpy

```
import numpy as np
```

Kreiranje numpy niza

- NumPy najvećim delom radi sa objektima tipa ndarray
- Ndarray je višedimenzionalni niz
- Generisanje ndarray objekta, vrši se pozivom funkcije array
- Funkciji array prosleđuje se nabrojivi tip ili se niz generiše funkcijom **arange** ili transformacijom python sekvence

```
arr = np.array([1,2,3,4])
```

```
arr = np.arange(1,5)
```

- Dobijeni objekat je tipiziran i tretiran kao matrica
- Svaki niz (red) matrice, naziva se osa **(axis)**

```
[1 2 3 4]
```

Tipizacija numpy niza

- Numpy nizovi su strogo tipizirani
- Tip niza se određuje automatski

```
arr = np.array([1,2,3,4])
```

Tip je **int64**

```
arr = np.array([1.,2,3,4])
```

Tip je **float64**

```
arr = np.array(
    [0,1,2,3],
    dtype=np.bool)
```

Tip je **bool**

Dostupni tipovi podataka

https://numpy.org/devdocs/user/basics.types.html

Numpy type	Description
np.int8	Byte (-128 to 127)
np.int16	Integer (-32768 to 32767)
np.int32	Integer (-2147483648 to 2147483647)
np.int64	Integer (-9223372036854775808 to 9223372036854775807)
np.uint8	Unsigned integer (0 to 255)
np.uint16	Unsigned integer (0 to 65535)
np.uint32	Unsigned integer (0 to 4294967295)
np.uint64	Unsigned integer (0 to 18446744073709551615)
np.intp	Integer used for indexing, typically the same as ssize_t
np.uintp	Integer large enough to hold a pointer
np.float32	Note that this matches the precision of the builtin python float
np.float64 / np.float_	
np.complex64	Complex number, represented by two 32-bit floats (real and imaginary components)
np.complex128 / np.complex_	

Višedimenzionalni nizovi

- Nympe je u stanju da detektuje višedimenzionalne nizove, i podržava različite aritmetičke operacije nad njima

```
arr = np.array([[1,2,3],[4,5,6]])


print(arr)
```

```
[[1 2 3]
 [4 5 6]]
```

- Nympy je u stanju da detektuje višedimenzionalne nizove, i podržava različite aritmetičke operacije nad njima

```
print(arr*2)
```

```
[[ 2  4  6]
 [ 8 10 12]]
```

```
print(arr+2)
```

```
[[3 4 5]
 [6 7 8]]
```

```
print( np.array([1,2,3])+
    np.array([2,3,4]))
```

```
[3 5 7]
```

Aritmetika nad nizovima

- Numpy primenjuje aritmetičke operacije nad nizovima ukoliko se dimenzije mogu poklopiti:

```
a = np.array([[1,2],[3,4]])
b = np.array([[2,3],[4,5]])
print(a*b)
```

1*2 , 2*3
3*4 , 4*5

```
[[ 2  6]
 [12 20]]
```

Aritmetika nad nizovima

- Ukoliko se dimenzije ne poklapaju, numpy prijavljuje grešku

```
a = np.array([1,2,3])
b = np.array([1,2])
print(a*b)
```

1*1 , 2*2 , 3*?

```
ValueError: operands could not be broadcast together with shapes (3,) (2,)
```

Preoblikovanje nizova

- Numpy može preoblikovati višedimenzionalne nizove i tada se takođe mora voditi računa o tome da ciljni oblik odgovara trenutnom sadržaju

Može

```
arr = np.array([[1,2,3,4],[5,6,7,8]])
arr = arr.reshape(4,2)
print(arr)
```

```
[[1 2]
 [3 4]
 [5 6]
 [7 8]]
```

Ne može

```
arr = np.array([[1,2,3,4],[5,6,7,8]])
arr = arr.reshape(3,2)
print(arr)
```

```
ValueError: cannot reshape array of size 8 into shape (3,2)
```

Kompresovanje nizova

- Metod squeeze, svodi niz na vrednosti jednodimenzionalnih nizova

```
arr = [
    [
        [1],[2],[3]
    ]
]
print(np.squeeze(arr))
```

```
[1 2 3]
```

Transponovanje matrice

```
arr = np.array([
    [1,2,3,4,5,6,7,8,9,10],
    [1,2,3,4,5,6,7,8,9,10]
  ]
)
print(arr.T)
```

```
print(arr.T)
```

Matrična aritmetika nad nizovima

- Definisanje jedinične matrice

```
mat = np.eye(4)
```

```
mat = np.identity(4)
```

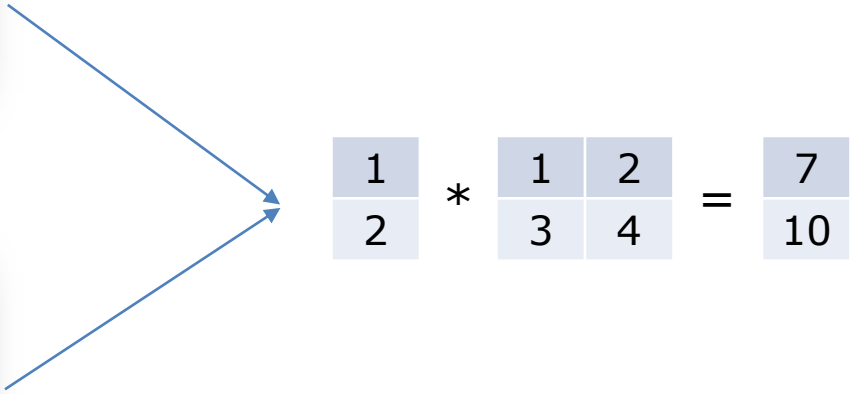
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

- Definisanje jedinične matrice

Množenje matrica

```
a = np.array([1,2])
b = np.array([[1,2],[3,4]])
c = np.matmul(a,b)
```

```
a = np.array([1,2])
b = np.array([[1,2],[3,4]])
c = a@b
print(c)
```



Struktuirani nizovi

- Numpy omogućava simuliranje strukture podataka kroz specijalno definisane višedimenzionalne nizove

```
tp=[('name', 'U10'), ('price', float)]

b = np.array([("Phone",125.99),("Bicycle",100.5),("TV",82.22)],tp)

print(
    b[0]["name"],
    b[0]["price"]
)
```